

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application. Please amend the claims as follows:

Listing of Claims:

1. (Currently Amended) A computer program product stored on at least one physical computer readable media and encoding a computer program for executing on a computer system a computer implemented method for managing allocation of program data in a target program between one or more thread-specific heaps and at least one shared heap, the program data including thread-specific data and shared data, the computer implemented method comprising:

analyzing the target program during code compilation to distinguish ~~between~~among the proven thread-specific data of a first program thread, the proven thread-specific data of a second program thread, and the shared data, wherein program data in the target program is identified as the shared data, if the program data is referenced by the first program thread and the second program thread;

configuring the target program to allocate the proven thread-specific data of the first program thread to a first thread-specific heap, responsive to the analyzing operation; configuring the target program to allocate the proven thread-specific data of the second program thread to a second thread-specific heap, responsive to the analyzing operation; and configuring the target program to allocate the shared data to the shared heap, responsive to the analyzing operation.

2. (Cancelled)

3. (Original) The computer program product of claim 1 wherein the analyzing operation comprises:

identifying program data in the target program as the thread-specific data of the first program thread, if the program data is not referenced by any other program thread of the target program.

4. (Original) The computer program product of claim 1 wherein the analyzing operation comprises:

identifying program data in the target program as the thread-specific data of the first program thread based on a thread escape analysis.

5. (Cancelled)

6. (Original) The computer program product of claim 1 wherein the analyzing operation occurs prior to execution of the target program.

7. (Currently Amended) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the proven thread-specific data of the first program thread comprises:

replacing an original allocation instruction in the target program with a new instruction that allocates the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread.

8. (Currently Amended) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the proven thread-specific data of the first program thread comprises:

leaving an original allocation instruction in the target program to allocate the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread.

9. (Original) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the shared data comprises:

leaving an original allocation instruction in the target program to allocate the shared data to the shared heap.

10. (Original) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the shared data comprises:

replacing an original allocation instruction in the target program with a new instruction that allocates the shared data to the shared heap.

11. (Currently Amended) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the proven thread-specific data of the first program thread comprises:

configuring an allocation parameter associated with the thread-specific data indicating that the thread-specific data of the first program thread is to be allocated in the one of the thread-specific heaps.

12. (Currently Amended) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the proven thread-specific data of the first program thread further comprises:

allocating the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread, responsive to an allocation parameter.

13. (Original) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the shared data comprises:

configuring an allocation parameter associated with the shared data indicating that the shared data is to be allocated in the shared heap.

14. (Original) The computer program product of claim 13 wherein the operation of configuring the target program to allocate the shared data further comprises:

allocating the shared data to the shared heap, responsive to the allocation parameter.

15. (Currently Amended) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the proven thread-specific data of the first program thread occurs prior to execution of the target program.

16. (Original) The computer program product of claim 1 wherein the operation of configuring the target program to allocate the shared data occurs prior to execution of the target program.

17. (Previously Presented) The computer program of claim 1 wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of garbage collection of the shared data in the shared heap.

18. (Currently Amended) The computer program of claim 1 wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of garbage collection of at the second thread-specific heap.

19. (Previously Presented) The computer program of claim 1 wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of the execution of another program thread in the target program.

20. (Previously Presented) The computer program of claim 1 wherein the computer implemented method further comprises:

garbage collecting the shared data from the shared heap independently of garbage collection of the thread-specific data in the first thread-specific heap.

21. (Previously Presented) The computer program of claim 1 wherein the computer implemented method further comprises:

maintaining a remembered set identifying references to one or more shared data in the shared heap; and

collecting the shared heap independently of garbage collection of the first thread-specific heap.

22. (Previously Presented) The computer program product of claim 1 wherein the computer implemented method further comprises:

collecting a portion of the shared data from the shared heap to leave an uncollected portion of the shared data in the shared heap, the uncollected portion of the shared data including shared data that is referenced by thread-specific data of the first thread-specific heap that has not yet been scanned;

scanning the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data; and

collecting the uncollected portion of the shared data from the shared heap, responsive to the scanning operation.

23. (Previously Presented) The computer program product of claim 22 wherein the computer implemented method further comprises:

collecting the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data.

24. (Original) The computer program product of claim 1 wherein the shared heap is shared by a subset of the program threads of the target program, wherein the subset of program threads includes less than all of the program threads of the target program.

25. (Currently Amended) A method implemented at least in part by a computing device for allocating of program data in a target program between one or more thread-specific heaps and at least one shared heap, the program data including thread-specific data and shared data, the method comprising:

analyzing the target program during code compilation to distinguish between proven thread-specific data of a first program thread and the shared data, wherein program data in the target program is identified as the shared data, if the program data is referenced by the first program thread and a second program thread of the target program;

configuring the target program to allocate the proven thread-specific data of the first program thread to a first thread-specific heap, responsive to the analyzing operation, by replacing an original allocation instruction in the target program with a new instruction that allocates the thread-specific data of the first program thread to the first thread-specific heap associated with the first program thread; and

configuring the target program to allocate the shared data to the shared heap, responsive to the analyzing operation, by replacing an original allocation instruction in the target program with a new instruction that allocates the shared data to the shared heap.

26. (Cancelled)

27. (Cancelled)

28. (Cancelled)

29. (Cancelled)

30. (Cancelled)

31. (Original) The method of claim 25 further comprising:

collecting a portion of the shared data from the shared heap to leave an uncollected portion of the shared data in the shared heap, the uncollected portion of the shared data including shared data that is referenced by thread-specific data of the first thread-specific heap that has not yet been scanned;

scanning the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data; and

collecting the uncollected portion of the shared data from the shared heap, responsive to the scanning operation.

32. (Original) The method of claim 31 further comprising:

collecting the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data.

33. (Original) The method of claim 25 wherein the operation of configuring the target program to allocate the thread-specific data comprises:

configuring an allocation parameter associated with the thread-specific data indicating that the thread-specific data of the first program thread is to be allocated in the first thread-specific heap.

34. (Currently Amended) A compiler implemented at least in part by a computing device for managing allocation of program data of a target program between a shared heap and a thread-specific heap, the program data including thread-specific data and shared data, the compiler comprising:

a program analyzer analyzing the target program during code compilation to distinguish between proven thread-specific data of a first program thread, proven thread-specific data of a second program thread, and the shared data, wherein program data in the target program is identified as the thread-specific data of the first program thread, if the program data is not referenced by any other program thread of the target program; and

a code specializer configuring the target program to allocate the proven thread-specific data of the first program thread to a first thread-specific heap, configuring the target program to allocate the thread-specific data of the second program thread to a second thread-specific heap, and configuring the target program to allocate the shared data to the shared heap, responsive to the analyzing operation.

35. (Currently Amended) A computer program product stored on at least one physical computer readable media and encoding a computer program for executing on a computer system a computer implemented method for managing memory used for program data in a target

program having one or more thread-specific heaps and at least one shared heap, the program data including thread-specific data and shared data, the computer implemented method comprising:

analyzing the target program during code compilation to distinguish between proven thread-specific data of a first program thread and the shared data;

allocating during target program code compilation the proven thread-specific data associated with a first program thread of the target program to a first thread-specific heap, the thread-specific data being determined to be reachable only by the first thread; and

allocating during target program code compilation the shared data to the shared heap, the shared data being deemed potentially reachable by a plurality of the program threads of the target program;

garbage collecting the thread-specific data from the first thread-specific heap independently of garbage collection of the shared data in the shared heap; and

garbage collecting the shared data from the shared heap independently of garbage collection of the thread-specific data in the first thread-specific heap.

36. (Cancelled)

37. (Previously Presented) The computer program of claim 35 wherein the computer implemented method further comprises:

garbage collecting the thread-specific data from the first thread-specific heap independently of the execution of another program thread in the target program.

38. (Cancelled)

39. (Previously Presented) The computer program of claim 35 wherein the computer implemented method further comprises:

maintaining a remembered set identifying references to one or more shared data in the shared heap; and

collecting the shared heap independently of garbage collection of the first thread-specific heap, based on the references identified in the remembered set.

40. (Currently Amended) A method implemented at least in part by a computing device for managing memory used for program data in a target program having one or more thread-specific heaps and at least one shared heap, the program data including thread-specific data and shared data, the method comprising:

analyzing the target program during code compilation to distinguish between proven thread-specific data of a first program thread and the shared data;

allocating the proven thread-specific data associated with a first program thread of the target program during code compilation to a first thread-specific heap, the thread-specific data being determined to be reachable only by the first thread; and

allocating the shared data to the shared heap during code compilation, the shared data being deemed potentially reachable by a plurality of the program threads of the target program;

garbage collecting the thread-specific data from the first thread-specific heap independently of the execution of another program thread in the target program; and

garbage collecting the shared data from the shared heap independently of garbage collection of the thread-specific data in the first thread-specific heap.

41. (Original) The method of claim 40 further comprising:

garbage collecting the thread-specific data from the first thread-specific heap

independently of garbage collection of the shared data in the shared heap.

42. (Cancelled)

43. (Cancelled)

44. (Original) The method of claim 40 further comprising:

maintaining a remembered set identifying references to one or more shared data in the shared heap; and

collecting the shared heap independently of garbage collection of the first thread-specific heap, based on the remembered set.

45. (Original) The method of claim 40 further comprising:

collecting a portion of the shared data from the shared heap to leave an uncollected portion of the shared data in the shared heap, the uncollected portion of the shared data including shared data that is referenced by thread-specific data of the first thread-specific heap that has not yet been scanned;

scanning the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data; and

collecting the uncollected portion of the shared data from the shared heap, responsive to the scanning operation.

46. (Original) The method of claim 45 further comprising:

collecting the thread-specific data from the first thread-specific heap, responsive to the operation of collecting a portion of the shared data.

47. (Currently Amended) A memory manager implemented at least in part by a computing device for managing heap memory in a computer system, the heap memory being used to store program data, the program data including thread-specific data and shared data, the memory manager comprising:

a program analyzer analyzing the target program during code compilation to distinguish between proven thread-specific data of a first program thread and the shared data; and

an allocation module allocating the proven thread-specific data associated with the first program thread of the target program to a first thread-specific heap, the thread-specific data being determined to be reachable only by the first thread, and allocating the shared data to the shared heap, the shared data being deemed potentially reachable by a plurality of the program threads of the target program; and

a garbage collector reclaiming memory associated with the thread-specific data from the first thread-specific heap independently of garbage collection of the shared data in the shared heap, and independently of the execution of another program thread in the target program.

48. (Cancelled)

49. (Cancelled)

50. (Original) The memory manager of claim 47 further comprising:

a garbage collector reclaiming memory associated with the shared data from the shared heap independently of garbage collection of the thread-specific data in the first thread-specific heap.

51. (Original) The memory manager of claim 47 wherein the memory manager maintains a remembered set identifying references to one or more shared data in the shared heap and further comprising:

a garbage collector reclaiming memory associated with the shared heap independently of garbage collection of the first thread-specific heap, based on the remembered set.